

Docker

Quase tudo o que você precisa saber sobre gerenciamento de containers

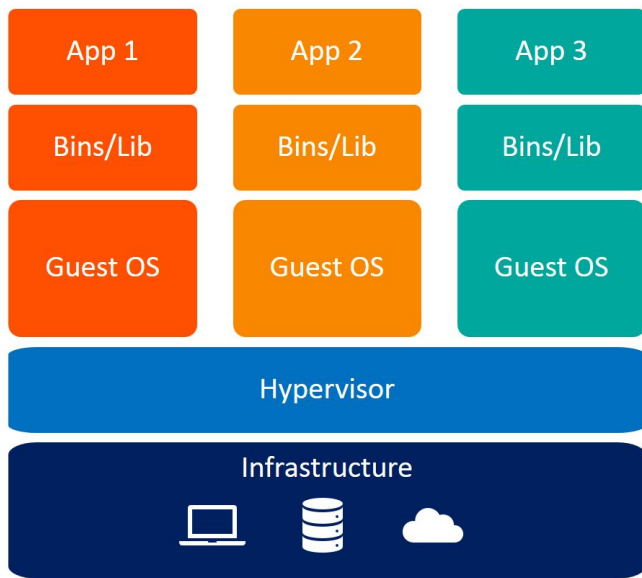
Docker

- Enterprise Container Platform
- Rodar aplicações em um ambiente isolado
- Baseado em containers. Um pouquinho diferente de virtualização.
- Funciona em qualquer lugar, basta ter uma imagem base.
- Sandbox. Evita conflitos entre dependências e erros de environment.
Aumenta segurança.

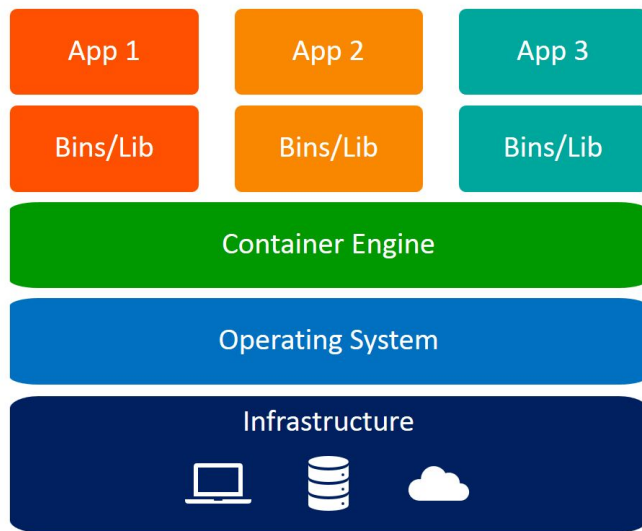
Containers

- Instância (em execução) de uma imagem.
- Não possui necessidade de hypervisor.
 - Hypervisor: Controla o acesso do SO virtualizado a recursos de hardware.
- Diminuem abstração.
- Não fazem build de um sistema operacional novo.

Containers



Virtual Machines



Containers

Imagem

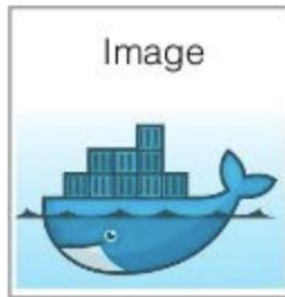
- Template de criação do ambiente (snapshot).
- Especifica:
 - Sistema operacional.
 - Dependências.
 - Aplicação.
- Dockerfile: configura uma imagem.
- Imagens podem derivar de outras.
- Docker Hub.

Imagens

```
FROM ubuntu:16.04
MAINTAINER John Doe <john.doe@example.com>
WORKDIR /app
COPY . /app
RUN apt-get update && apt-get install -y python3
RUN python3 --help
```

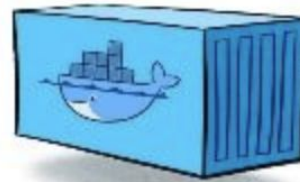
Dockerfile

build



Docker Image

run



Docker Container

Microservices

- Docker Compose.
 - Facilita sua vida (e muito!).
- Voltado para aplicações com múltiplos componentes presentes em diferentes containers.
- `docker-compose.yml`

Clusters

- Docker Swarm.
- Gerenciar containers rodando em diferentes hosts.
- Alternativa: Kubernetes (Google)
 - Pokemon GO usa (<http://bit.do/kubernetes-pokemon>).

Hands on

Lembretes e Dicas

- Containers foram criados para componentes stateless. Evitem incluir componentes stateful (i.e. banco, storage).
- O uso de volumes pode resolver esse problema (porém não garante integridade).
- Usar screens é muito melhor do que ficar rodando “docker run -it” para ver os recursos da máquina.
- Que tal termos nosso próprio Docker Hub?

FIM!

(foi mal a demora para
apresentar!)